



Inferencing and beyond: further adventures with parity-based architectures for decentralized discrete-event systems

Laurie Ricker, T F Lidbetter, Hervé Marchand

► To cite this version:

Laurie Ricker, T F Lidbetter, Hervé Marchand. Inferencing and beyond: further adventures with parity-based architectures for decentralized discrete-event systems. 20th World Congress The International Federation of Automatic Control, Jul 2017, Toulouse, France. pp.6. hal-01590438

HAL Id: hal-01590438

<https://inria.hal.science/hal-01590438>

Submitted on 19 Sep 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Inferencing and beyond: further adventures with parity-based architectures for decentralized discrete-event systems

S.L. Ricker * T.F. Lidbetter * H. Marchand **

* *Dept. of Mathematics & Computer Science,
Mount Allison University
Sackville, NB, Canada (e-mail: {lricker, tfidbetter}@mta.ca)*
** *INRIA Rennes-Bretagne Atlantique,
Rennes, France (e-mail: herve.marchand@inria.fr)*

Abstract: We extend our examination of decentralized discrete-event systems architectures that use *exclusive or* (XOR) as the fusion rule to reach control decisions. A characterization of XOR inference-observable languages is provided. Additionally, XOR observability is defined for languages that are not inference observable but are distributed observable.

1. INTRODUCTION

Decentralized decision-making architectures for discrete-event systems are characterized by decentralized agents that take *local* decisions, based on their own partial observations, and reach a *global* decision by fusing the local decisions using logical operators, such as conjunction and disjunction (Rudie and Wonham, 1992; Prosser et al., 1997; Yoo and Lafortune, 2002) and, sometimes, a combination of the two (Chakib and Khoumsi, 2011).

We previously introduced the idea of incorporating XOR as a fusion rule for decentralized discrete-event systems (Ricker and Marchand, 2013). The problem of finding a correct control solution was reduced to that of solving XORSAT (Biere et al., 2009), although this only provided a necessary condition for the existence of an XOR solution. Notably, the decentralized architecture of Qiu et al. (2008) also uses XOR as a fusion rule, but to respect parity, they insist that exactly one controller takes the correct control decision. In general, our approach requires no such restriction.

We extend our results of Ricker and Marchand (2013) by providing language-based characterizations for when a parity-based architecture exists. In particular, we examine XOR as a fusion rule for

- languages that are inference observable (Kumar and Takai, 2007); and,
- languages that are distributed observable (Ricker and Rudie, 2000) but not inference observable.

2. BACKGROUND

We assume the framework for supervisory control of discrete-event systems as introduced in Ramadge and Wonham (1987). Accordingly, we model a discrete-event system as a finite-state automaton

$$M_L = (Q, \Sigma, T_L, q_0),$$

where Q is a finite set of states, Σ is a finite alphabet of events, $T_L \subseteq Q \times \Sigma \times Q$ is a transition relation,

and $q_0 \in Q$ is the initial state. In addition, Σ is the disjoint union of two types of events: (i) observable and unobservable events $\Sigma = \Sigma_o \uplus \Sigma_{uo}$; and (ii) controllable and uncontrollable events $\Sigma = \Sigma_c \uplus \Sigma_{uc}$. The transition relation is easily extended to Σ^* and we say $L := \{s \in \Sigma^* \mid \exists q \in Q \text{ s.t. } q_0 \xrightarrow{s} q \in T_L\}$. For $L \subseteq \Sigma^*$, we have $\bar{L} := \{v \in \Sigma^* \mid \exists w \in \Sigma^*, u \in L \text{ such that } u = vw\}$. Then L is *prefix-closed* if $L = \bar{L}$. We assume prefix-closed languages for the remainder of this paper.

Given a description of the behavior of the uncontrolled system as L , the regular language generated by M_L , which describes the behavior of the uncontrolled system, and a specification language $K \subseteq L$, generated by M_K , the object of the control problem is to prevent sequences in $L \setminus K$ from occurring. In a decentralized framework, where there is a set of $n > 1$ controllers, denoted by $I = \{1, \dots, n\}$, this amounts to the existence of at least one decentralized controller, based only on its observation of L , to be able to definitively identify when a sequence in $L \setminus K$ is about to occur (Rudie and Wonham, 1992), i.e., issues a “disable” command for the controllable event that takes the system from K into $L \setminus K$.

Each decentralized controller $i \in I$ has a set of events that it observes, $\Sigma_{o,i} \subseteq \Sigma_o$, and a set of events it controls, $\Sigma_{c,i} \subseteq \Sigma_c$. We let $I_c(\sigma) = \{i \in I \mid \sigma \in \Sigma_{c,i}\}$. The *natural projection* $\pi_i : \Sigma^* \rightarrow \Sigma_{o,i}^*$ defines the observations of an agent $i \in I$ by removing all occurrences of events in $\Sigma \setminus \Sigma_{o,i}$ from a sequence $s \in \Sigma^*$. Furthermore, we will refer to the natural projection w.r.t. Σ_o by π , namely, $\pi : \Sigma^* \rightarrow \Sigma_o^*$. The inverse projection $\pi_i^{-1} : \Sigma_{o,i}^* \rightarrow 2^{\Sigma^*}$ captures all the sequences $s' \in \Sigma^*$ that produce the same natural projection for agent i as s does. We will use the notation $\llbracket s \rrbracket_i$ whenever we refer to $\pi_i^{-1} \pi_i(s) \cap L$.

Our goal is to define, for each controller $i \in I$, a local decision function h_i over its partial observation of L such that the fusion of the local decisions, using a fusion rule such as \vee or \wedge , reaches the correct global control, as indicated by the specification. There are two decentralized

architectures to consider: (i) *enable by default* (EBD), where uncertainty on the part of a decentralized controller means the local decision is “enable”, represented by 0, and uses \vee as a fusion rule (Rudie and Wonham, 1992); (ii) *disable by default* (DBD), where uncertainty on the part of a decentralized controller means the local decision is “disable”, represented by 1, and uses \wedge as the fusion rule (Yoo and Lafortune, 2002).

Definition 1. Given regular languages L, K where $K \subseteq L \subset \Sigma^*$, n local decision functions $h_i : \Sigma_{o,i}^* \rightarrow \{0, 1\}^\Sigma$ (for $i \in I$) and the global decision function $H : \Sigma_o \rightarrow \{0, 1\}^\Sigma$, such that $H(\pi(t))(\sigma) = \odot_{i \in I} h_i(\pi_i(t))$ for $t \in L$ and $\sigma \in \Sigma$, where \odot is a placeholder for the fusion rule. Then H forms a **valid** global decision function w.r.t. fusion rule \odot whenever

$$\begin{aligned} (\forall t \in K)(\forall \sigma \in \Sigma) t\sigma \in K &\Rightarrow H(\pi(t))(\sigma) = 0 \text{ and} \\ (\forall t \in K)(\forall \sigma \in \Sigma_{uc}) H(\pi(t))(\sigma) &= 0, \\ (\forall t \in K)(\forall \sigma \in \Sigma_c) t\sigma \in L \setminus K &\Rightarrow H(\pi(t))(\sigma) = 1. \end{aligned}$$

We can find such functions if prefix-closed K is (i) controllable; and (ii) inference observable (Rudie and Wonham, 1992; Kumar and Takai, 2007).

The language generated by M_L under the control of the global decision function H w.r.t. a given fusion rule is denoted by $\mathcal{L}(H/M_L)$, and is the largest language such that $\mathcal{L}(H/M_L) \subseteq \mathcal{L}(M_L)$ where

- $\varepsilon \in \mathcal{L}(H/M_L)$; and
- $(\forall t \in \mathcal{L}(H/M_L))(\forall \sigma \in \Sigma)$
 $t\sigma \in L \wedge H(\pi(t))(\sigma) = 0 \Leftrightarrow t\sigma \in \mathcal{L}(H/M_L)$.

Definition 2. (Ramadge and Wonham (1987)) A language $K \subseteq L$ is **controllable** w.r.t. L and Σ_{uc} if $K\Sigma_{uc} \cap L \subseteq K$.

The definitions below are simply a rewriting of those in Kumar and Takai (2007) and classified by their decision-making architecture.¹

Definition 3. A language $K \subseteq L$ is **EBD inference observable** w.r.t. L , π_i and $\Sigma_{c,i}$ (for $i \in I$) with fusion rule \vee if either

$$\begin{aligned} (\forall t \in K)(\forall \sigma \in \Sigma) t\sigma \in L \setminus K &\Rightarrow \\ (\exists m \in \mathbb{N})(\forall i_0 \in I_c(\sigma)) \llbracket t \rrbracket_{i_0} \sigma \cap K &\neq \emptyset \Rightarrow \\ (\forall i_1 \in I_c(\sigma) \setminus \{i_0\})(\forall t_1 \sigma \in \llbracket t \rrbracket_{i_0} \sigma \cap K) \llbracket t_1 \rrbracket_{i_1} \sigma \cap L \setminus K &\neq \emptyset \Rightarrow \\ (\forall i_2 \in I_c(\sigma) \setminus \{i_1\})(\forall t_2 \sigma \in \llbracket t_1 \rrbracket_{i_1} \sigma \cap L \setminus K) \llbracket t_2 \rrbracket_{i_2} \sigma \cap K &\neq \emptyset \Rightarrow \\ \dots & \end{aligned} \quad (1)$$

$$\begin{aligned} (\forall i_{m-1} \in I_c(\sigma) \setminus \{i_{m-2}\})(\forall t_{m-1} \sigma \in \llbracket t_{m-2} \rrbracket_{i_{m-2}} \sigma \cap K) \\ \llbracket t_{m-1} \rrbracket_{i_{m-1}} \sigma \cap L \setminus K &\neq \emptyset \Rightarrow \\ (\exists i'_{m-1} \in I_c(\sigma) \setminus \{i_{m-2}\})(\forall t_{m-1} \sigma \in \llbracket t_{m-1} \rrbracket_{i'_{m-1}} \sigma \cap L \setminus K) \\ (\exists i_m \in I_c(\sigma) \setminus \{i_{m-1}\}) \llbracket t_m \rrbracket_{i_m} \sigma \cap K &= \emptyset. \end{aligned}$$

or

$$\begin{aligned} (\forall t \in K)(\forall \sigma \in \Sigma) t\sigma \in K &\Rightarrow \\ (\exists m \in \mathbb{N})(\forall i_0 \in I_c(\sigma)) \llbracket t \rrbracket_{i_0} \sigma \cap L \setminus K &\neq \emptyset \Rightarrow \\ (\forall i_1 \in I_c(\sigma) \setminus \{i_0\})(\forall t_1 \sigma \in \llbracket t \rrbracket_{i_0} \sigma \cap L \setminus K) \llbracket t_1 \rrbracket_{i_1} \sigma \cap K &\neq \emptyset \Rightarrow \\ (\forall i_2 \in I_c(\sigma) \setminus \{i_1\})(\forall t_2 \sigma \in \llbracket t_1 \rrbracket_{i_1} \sigma \cap K) \llbracket t_2 \rrbracket_{i_2} \sigma \cap L \setminus K &\neq \emptyset \Rightarrow \\ \dots & \end{aligned} \quad (2)$$

$$\begin{aligned} (\forall i_{m-1} \in I_c(\sigma) \setminus \{i_{m-2}\})(\forall t_{m-1} \sigma \in \llbracket t_{m-2} \rrbracket_{i_{m-2}} \sigma \cap K) \\ \llbracket t_{m-1} \rrbracket_{i_{m-1}} \sigma \cap L \setminus K &\neq \emptyset \Rightarrow \\ (\exists i'_{m-1} \in I_c(\sigma) \setminus \{i_{m-2}\})(\forall t_{m-1} \sigma \in \llbracket t_{m-1} \rrbracket_{i'_{m-1}} \sigma \cap L \setminus K) \\ (\exists i_m \in I_c(\sigma) \setminus \{i_{m-1}\}) \llbracket t_m \rrbracket_{i_m} \sigma \cap K &= \emptyset. \end{aligned}$$

¹ In the sequel, we use the color red to distinguish sequences $t \in K$ such that $t\sigma \in L \setminus K$.

Decentralized controllers use the EBD architecture if the inferencing (eventually) leads to (at least) one of the controllers in $I_c(\sigma)$ having no ambiguity about particular subsets of *disablement* decisions for σ .

Definition 4. A language $K \subseteq L$ is **DBD inference observable** w.r.t. L , π_i and $\Sigma_{c,i}$ (for $i \in I$) with fusion rule \wedge if either

$$\begin{aligned} (\forall t \in K)(\forall \sigma \in \Sigma) t\sigma \in L \setminus K &\Rightarrow \\ (\exists m \in \mathbb{N})(\forall i_0 \in I_c(\sigma)) \llbracket t \rrbracket_{i_0} \sigma \cap K &\neq \emptyset \Rightarrow \\ (\forall i_1 \in I_c(\sigma) \setminus \{i_0\})(\forall t_1 \sigma \in \llbracket t \rrbracket_{i_0} \sigma \cap K) \llbracket t_1 \rrbracket_{i_1} \sigma \cap L \setminus K &\neq \emptyset \Rightarrow \\ (\forall i_2 \in I_c(\sigma) \setminus \{i_1\})(\forall t_2 \sigma \in \llbracket t_1 \rrbracket_{i_1} \sigma \cap L \setminus K) \llbracket t_2 \rrbracket_{i_2} \sigma \cap K &\neq \emptyset \Rightarrow \\ \dots & \end{aligned} \quad (3)$$

$$\begin{aligned} (\forall i_{m-1} \in I_c(\sigma) \setminus \{i_{m-2}\})(\forall t_{m-1} \sigma \in \llbracket t_{m-2} \rrbracket_{i_{m-2}} \sigma \cap L \setminus K) \\ \llbracket t_{m-1} \rrbracket_{i_{m-1}} \sigma \cap K &\neq \emptyset \Rightarrow \\ (\exists i'_{m-1} \in I_c(\sigma) \setminus \{i_{m-2}\})(\forall t_{m-1} \sigma \in \llbracket t_{m-1} \rrbracket_{i'_{m-1}} \sigma \cap K) \\ (\exists i_m \in I_c(\sigma) \setminus \{i_{m-1}\}) \llbracket t_m \rrbracket_{i_m} \sigma \cap L \setminus K &= \emptyset. \end{aligned}$$

or

$$\begin{aligned} (\forall t \in K)(\forall \sigma \in \Sigma) t\sigma \in K &\Rightarrow \\ (\exists m \in \mathbb{N})(\forall i_0 \in I_c(\sigma)) \llbracket t \rrbracket_{i_0} \sigma \cap L \setminus K &\neq \emptyset \Rightarrow \\ (\forall i_1 \in I_c(\sigma) \setminus \{i_0\})(\forall t_1 \sigma \in \llbracket t \rrbracket_{i_0} \sigma \cap L \setminus K) \llbracket t_1 \rrbracket_{i_1} \sigma \cap K &\neq \emptyset \Rightarrow \\ (\forall i_2 \in I_c(\sigma) \setminus \{i_1\})(\forall t_2 \sigma \in \llbracket t_1 \rrbracket_{i_1} \sigma \cap K) \llbracket t_2 \rrbracket_{i_2} \sigma \cap L \setminus K &\neq \emptyset \Rightarrow \\ \dots & \end{aligned} \quad (4)$$

$$\begin{aligned} (\forall i_{m-1} \in I_c(\sigma) \setminus \{i_{m-2}\})(\forall t_{m-1} \sigma \in \llbracket t_{m-2} \rrbracket_{i_{m-2}} \sigma \cap L \setminus K) \\ \llbracket t_{m-1} \rrbracket_{i_{m-1}} \sigma \cap K &\neq \emptyset \Rightarrow \\ (\exists i'_{m-1} \in I_c(\sigma) \setminus \{i_{m-2}\})(\forall t_{m-1} \sigma \in \llbracket t_{m-1} \rrbracket_{i'_{m-1}} \sigma \cap K) \\ (\exists i_m \in I_c(\sigma) \setminus \{i_{m-1}\}) \llbracket t_m \rrbracket_{i_m} \sigma \cap L \setminus K &= \emptyset. \end{aligned}$$

Decentralized controllers use the DBD architecture if the inferencing (eventually) leads to (at least) one of the controllers in $I_c(\sigma)$ having no ambiguity about particular subsets of *enablement* decisions for σ . When relevant, we will distinguish EBD from DBD inference-observability, otherwise we will simply refer to K as being inference observable.

It will be useful to discuss inferencing performed for a specific sequence $t\sigma \in L$.

Definition 5. An **inferencing chain** for sequence $t\sigma \in L$ is of two types and is characterized by the following progression of estimates for some sequence of indices $(i_m)_{m \in \mathbb{N}}$, where $i_0, \dots, i_m \in I$, as follows.

(a) **EBD inferencing chains:**

$$\begin{aligned} \llbracket t_m \rrbracket \in [\dots \llbracket t_2 \rrbracket \in [\llbracket t_1 \rrbracket \in [\llbracket t \rrbracket_{i_0} \sigma \cap K]_{i_1} \sigma \cap L \setminus K]_{i_2} \sigma \dots]_{i_{m-1}} \sigma \cap L \setminus K]_{i_m} \sigma \cap K &= \emptyset; \\ \llbracket t_m \rrbracket \in [\dots \llbracket t_2 \rrbracket \in [\llbracket t_1 \rrbracket \in [\llbracket t \rrbracket_{i_0} \sigma \cap L \setminus K]_{i_1} \sigma \cap K]_{i_2} \sigma \dots]_{i_{m-1}} \sigma \cap L \setminus K]_{i_m} \sigma \cap K &= \emptyset; \end{aligned}$$

(b) **DBD inferencing chains**

$$\begin{aligned} \llbracket t_m \rrbracket \in [\dots \llbracket t_2 \rrbracket \in [\llbracket t_1 \rrbracket \in [\llbracket t \rrbracket_{i_0} \sigma \cap K]_{i_1} \sigma \cap L \setminus K]_{i_2} \sigma \dots]_{i_{m-1}} \sigma \cap K]_{i_m} \sigma \cap L \setminus K &= \emptyset; \\ \llbracket t_m \rrbracket \in [\dots \llbracket t_2 \rrbracket \in [\llbracket t_1 \rrbracket \in [\llbracket t \rrbracket_{i_0} \sigma \cap L \setminus K]_{i_1} \sigma \cap K]_{i_2} \sigma \dots]_{i_{m-1}} \sigma \cap K]_{i_m} \sigma \cap L \setminus K &= \emptyset. \end{aligned}$$

For brevity, and when clear from the context, we will denote the result of the complete inferencing chain for $t \in K$ and $\sigma \in \Sigma_c$ w.r.t. $i_m \in I_c(\sigma)$ by $\llbracket t \rrbracket_{i_m} \sigma$. Occasionally, we will refer to the k th link in an inferencing chain for $t\sigma$ as $\llbracket t_k \rrbracket_{i_k} \sigma$, for $i_k \in (i_m)_{m \in \mathbb{N}}$ and $k \leq m$.

Finally, we revisit a decentralized language family from Ricker and Rudie (2000).

Definition 6. A language $K \subseteq L$ is **distributed observable** w.r.t. L , π_i and $\Sigma_{c,i}$ (for $i \in I$) if

$$(\forall t \in K)(\forall \sigma \in \Sigma_c) t\sigma \in L \setminus K \Rightarrow \cap_{i \in I_c(\sigma)} \llbracket t \rrbracket_i \sigma \cap K = \emptyset.$$

As noted in Ricker and Marchand (2013), the family of distributed-observable languages lies between the family of inference-observable languages and the family of observable languages (i.e., $n = 1$).

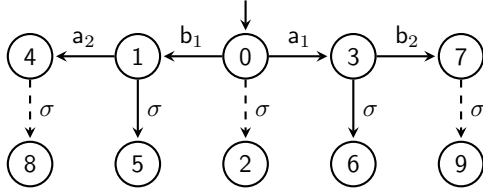


Fig. 1. Ex. 1 (from Yoo and Lafortune (2004)): $\Sigma_{o,1} = \{a_1, a_2\}$, $\Sigma_{o,2} = \{b_1, b_2\}$; $I_c(\sigma) = \{1, 2\}$.

Example 1. Fig. 1 features an inference-observable language K (only those sequences composed of entirely solid-line transitions) w.r.t. L (all sequences).

As all the inference chains for $t\sigma \in K$ are EBD inference chains, as shown in Table 1, K is EBD-inference observable. It is also the case that all the inference chains for $t\sigma \in L \setminus K$ are also EBD inference chains. Thus, we can use \vee as the overall fusion rule for all the control decisions. \diamond

Table 1. Links of inferencing chains for enablement decisions of Ex. 1.

$\llbracket a_1 \rrbracket_1 \sigma = \{a_1 \sigma, a_1 b_2 \sigma\}$	$\llbracket a_1 \rrbracket_2 \sigma = \{a_1 \sigma, \sigma\}$
$\llbracket a_1 b_2 \rrbracket_2 \sigma = \{a_1 b_2 \sigma\}$	$\llbracket \varepsilon \rrbracket_1 \sigma = \{\sigma, b_1 \sigma\}$
	$\llbracket b_1 \rrbracket_2 \sigma = \{b_1 \sigma, b_1 a_2 \sigma\}$
	$\llbracket b_1 a_2 \rrbracket_1 \sigma = \{b_1 a_2 \sigma\}$
$\llbracket b_1 \rrbracket_1 \sigma = \{b_1 \sigma, \sigma\}$	$\llbracket b_1 \rrbracket_2 \sigma = \{b_1 \sigma, b_1 a_2 \sigma\}$
$\llbracket \varepsilon \rrbracket_2 \sigma = \{\sigma, a_1 \sigma\}$	$\llbracket b_1 a_2 \rrbracket_1 \sigma = \{b_1 a_2 \sigma\}$
$\llbracket a_1 \rrbracket_1 \sigma = \{a_1, a_1 b_2 \sigma\}$	
$\llbracket a_1 b_2 \rrbracket_2 \sigma = \{a_1 b_2 \sigma\}$	

3. XOR INFERENCE-OBSERVABILITY

We are interested in examining a new property XOR inference-observability for an arbitrary number of decision-makers in $I_c(\sigma)$. Due to space constraints, we present analysis only for a pair of controllers $i, j \in I_c(\sigma)$, and assume that all other controllers simply issue 0 as their local decision. For reference, we include the truth table for XOR for $n = 2$ in Table 2.

Table 2. Truth tables for XOR (\oplus) for $n = 2$.

p	q	$p \oplus q$
0	0	0
0	1	1
1	0	1
1	1	0

We want to solve the following problem:

Problem 1. Given M_L, M_K that recognize prefix-closed L, K (such that $K \subseteq L$), a set of decentralized agents $I = \{1, \dots, n\}$, n sets of observable events $\Sigma_{o,1}, \dots, \Sigma_{o,n}$, and n sets of controllable events $\Sigma_{c,1}, \dots, \Sigma_{c,n}$. Find n local decision functions h_1, \dots, h_n , where $h_i : \Sigma_{o,i}^* \rightarrow \{0, 1\}^\Sigma$, that form a valid global decision function H w.r.t. fusion rule \oplus such that $\mathcal{L}(H/M_L) = K$, where 0 corresponds to enable and 1 corresponds to disable.

In Ricker and Marchand (2013) we noted a necessary condition for solving Problem 1: we cast the control problem as an instance of XOR-SAT (Biere et al., 2009) and if the resulting system of linear equations yielded a solution to XOR-SAT, we had a solution to Problem 1. Here, we introduce a language-based characterization of a necessary and sufficient condition to solve Problem 1.

When K is inference observable, we characterize when \oplus can be used as the fusion rule.

Definition 7. An inference-observable language $K \subseteq L$ is **XOR inference-observable** w.r.t. L, π_i and $\Sigma_{c,i}$ (for $i \in I$) if either

$$(\forall t \in K)(\forall \sigma \in \Sigma_c) t\sigma \in L \setminus K \Rightarrow (\exists m, m' \in \mathbb{N})$$

$$(\exists i_m, j_{m'} \in I_c(\sigma)) \llbracket t_m \rrbracket_{i_m} \sigma \cap \llbracket t_{m'} \rrbracket_{j_{m'}} \sigma \setminus \{t\sigma\} = \emptyset,$$

or

$$(\forall t \in K)(\forall \sigma \in \Sigma_c) t\sigma \in K \Rightarrow (\exists m, m' \in \mathbb{N})$$

$$(\exists i_m, j_{m'} \in I_c(\sigma)) \llbracket t_m \rrbracket_{i_m} \sigma \cap \llbracket t_{m'} \rrbracket_{j_{m'}} \sigma \setminus \{t\sigma\} = \emptyset,$$

where $\llbracket t_m \rrbracket_{i_m} \sigma$ and $\llbracket t_{m'} \rrbracket_{j_{m'}} \sigma$ are the last links in these two inference chains for t .

There is a special case when $m = m' = 0$. This is because $t\sigma \in \llbracket t_m \rrbracket_{i_m} \sigma$ and $t\sigma \in \llbracket t_{m'} \rrbracket_{j_{m'}} \sigma$, since $t_m = t_{m'} = t$. Thus, the intersection will not be empty; however, it is still possible to assign local control decisions to meet parity for \oplus . In this case, one controller can (arbitrarily) be assigned the disablement command, while the other takes the opposite decision. For this reason we remove $t\sigma$ from the intersection of the two inference chains.

Revisiting Example 1

We previously illustrated that K is inference observable and here we verify that K is XOR inference-observable. Here we apply Definition 7 to the enablement decisions w.r.t. σ . After consulting Table 1, we note the last links in the inference chains for $a\sigma$ (where $i_0 = 1, m = 1$ and $j_0 = 2, m' = 3$). Specifically, $\llbracket a \rrbracket_{i_m} \sigma = \{a_1 b_2 \sigma\}$ and $\llbracket a \rrbracket_{j_{m'}} \sigma = \{b_1 a_2 \sigma\}$, which have an empty intersection. This is also the situation for the inference chains for $b\sigma$ (where $i_0 = 1, m = 3$ and $j_0 = 2, m' = 1$). The intersection of these links is likewise empty: $\llbracket b \rrbracket_{i_m} \sigma = \{a_1 b_2 \sigma\}$ and $\llbracket b \rrbracket_{j_{m'}} \sigma = \{b_1 a_2 \sigma\}$. Thus, we conclude that K is XOR inference-observable. A possible solution is shown in Table 3. \diamond

Table 3. Possible XOR solution for Ex. 1

	$h_1(\cdot)(\sigma)$	$h_2(\cdot)(\sigma)$	\oplus	$H(\cdot)(\sigma)$
b_1	1	1	0	0
a_1	0	0	0	0
ε	1	0	1	1
$b_1 a_2$	0	1	1	1
$a_1 b_2$	0	1	1	1

3.1 Local control decisions

To reach an XOR solution, we must identify appropriate local control decision-making to accommodate the parity rules. As a result, unlike local decisions made under fusion rules of \wedge or \vee , a controller may not be taking a local

control decision consistent with its observations. We assign local control decisions based on the relevant inference chains. To respect parity for the two decision-makers, $i, j \in I_c(\sigma)$, only one controller can issue a disablement command.

When an inference chain begins with $t\sigma \in L \setminus K$, we must have the controllers $i, j \in I_c(\sigma)$ that are involved in the inference chain correctly disable σ after t . To respect parity of XOR, a global enablement decision is taken only when the local control decisions of i and j are different. Note that if $|I_c(\sigma)| > 2$, then the controllers in $I \setminus \{i, j\}$ issue 0 as their local control decisions for all sequences involved in the inference chain for $t\sigma$. We use the following set of local control decisions for $i, j \in I_c(\sigma)$, where this pattern repeats until the end of the inferencing chain is reached:

$$\begin{aligned} h_i(\pi_i(t))(\sigma) &\in \{0, 1\} & h_j(\pi_j(t))(\sigma) &= 1 - h_i(\pi_i(t))(\sigma) \quad (5) \\ h_i(\pi_i(t_1))(\sigma) &= h_j(\pi_j(t_1))(\sigma) & h_j(\pi_j(t_1))(\sigma) &= h_i(\pi_i(t))(\sigma) \\ h_i(\pi_i(t_2))(\sigma) &= 1 - h_j(\pi_j(t_1))(\sigma) & h_j(\pi_j(t_2))(\sigma) &= 1 - h_i(\pi_i(t_2))(\sigma) \\ h_i(\pi_i(t_3))(\sigma) &= h_j(\pi_j(t_3))(\sigma) & h_j(\pi_j(t_3))(\sigma) &= h_i(\pi_i(t_2))(\sigma) \\ h_i(\pi_i(t_4))(\sigma) &= 1 - h_j(\pi_j(t_3))(\sigma) & h_j(\pi_j(t_4))(\sigma) &= h_i(\pi_i(t_4))(\sigma) \end{aligned}$$

When an inference chain begins with $t\sigma \in K$, we must have the relevant controllers $i, j \in I_c(\sigma)$ correctly enable σ after t . To respect parity of XOR, a global enablement decision is taken only when the local control decisions are the same. As noted above, controllers in $I \setminus \{i, j\}$ issue 0 as their local control decisions for all sequences involved in the inference chain for $t\sigma$. We use the following set of local control decisions for $i, j \in I_c(\sigma)$, where this pattern repeats until the end of the inferencing chain is reached:

$$\begin{aligned} h_i(\pi_i(t))(\sigma) &\in \{0, 1\} & h_j(\pi_j(t))(\sigma) &= h_i(\pi_i(t))(\sigma) \quad (6) \\ h_i(\pi_i(t_1))(\sigma) &= 1 - h_j(\pi_j(t_1))(\sigma) & h_j(\pi_j(t_1))(\sigma) &= 1 - h_i(\pi_i(t))(\sigma) \\ h_i(\pi_i(t_2))(\sigma) &= h_j(\pi_j(t_1))(\sigma) & h_j(\pi_j(t_2))(\sigma) &= h_i(\pi_i(t_2))(\sigma) \\ h_i(\pi_i(t_3))(\sigma) &= 1 - h_j(\pi_j(t_3))(\sigma) & h_j(\pi_j(t_3))(\sigma) &= 1 - h_i(\pi_i(t_2))(\sigma) \\ h_i(\pi_i(t_4))(\sigma) &= h_j(\pi_j(t_3))(\sigma) & h_j(\pi_j(t_4))(\sigma) &= 1 - h_i(\pi_i(t_4))(\sigma) \end{aligned}$$

Revisiting Example 1 for the last time: We examine one of the inference chains for $t = a_1$, where $i = 2$ and $j = 1$. In this case, $t_1 = \varepsilon$, $t_2 = b_1$, and $t_3 = b_1a_2$. Following the local control decisions of Eq. (6), as $a_1\sigma \in K$, since no prior decisions have been assigned, let $h_2(\pi_1(a_1))(\sigma) = 0$. As an immediate consequence, $h_1(\pi_1(a_1))(\sigma) = 0$ and $h_1(\pi_1(\varepsilon))(\sigma) = 1$. Subsequently, $h_2(\pi_2(\varepsilon))(\sigma) = 0$. We then have $h_2(\pi_2(b_1))(\sigma) = 1$ and $h_1(\pi_1(b_1a_2))(\sigma) = 0$. Consequently, $h_1(\pi_1(b_1))(\sigma) = 1$ and $h_2(\pi_2(b_1a_2))(\sigma) = 1$. These values confirm the XOR solution presented in Table 3.

3.2 Inference observable but not XOR inference-observable

When K is inference observable but not XOR inference-observable, there is a sequence $t \in K$, $\sigma \in \Sigma_c$ and $i, j \in I_c(\sigma)$ for which the relevant inference chains have a non-empty intersection. We present such an example below.

Example 2. Let M_L and M_K be given as in Fig. 2. In this example, K is EBD inference-observable but is not XOR inference-observable.

Table 4 contains the inference chains for the disablement decisions of Ex. 2. In particular, we choose to apply Definition 7 to the inference chains for $t\sigma = a_1b_2\sigma$. Here we

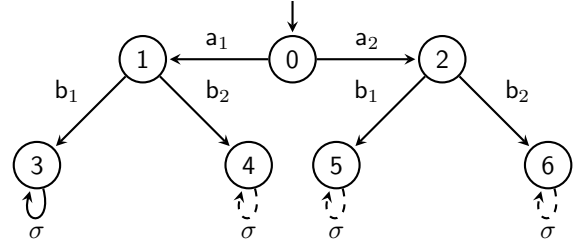


Fig. 2. Ex. 2: $\Sigma_{o,1} = \{a_1, a_2\}$, $\Sigma_{o,2} = \{b_1, b_2\}$; $I_c(\sigma) = \{1, 2\}$

have $\{a_2b_1\sigma, a_2b_2\sigma\} \cap \{a_1b_2\sigma, a_2b_2\sigma\} \setminus \{a_1b_2\sigma\} = \{a_2b_2\sigma\}$, and, therefore, conclude that K is not XOR-observable.

Table 4. Links of inferencing chains for enablement decisions of Ex. 2.

$[a_1b_2]_1\sigma = \{a_1b_2\sigma, a_1b_1\sigma\}$	$[a_1b_2]_2\sigma = \{a_1b_2\sigma, a_2b_2\sigma\}$
$[a_1b_1]_2\sigma = \{a_1b_1\sigma, a_2b_1\sigma\}$	
$[a_2b_1]_1\sigma = \{a_2b_1\sigma, a_2b_2\sigma\}$	
$[a_2b_1]_2\sigma = \{a_2b_1\sigma, a_1b_1\sigma\}$	
$[a_1b_1]_1\sigma = \{a_1b_1\sigma, a_1b_2\sigma\}$	
$[a_1a_2]_2\sigma = \{a_1a_2\sigma, a_2a_2\sigma\}$	
$[a_2b_2]_1\sigma = \{a_2b_1\sigma, a_2b_2\sigma\}$	$[a_2b_2]_2\sigma = \{a_1b_2\sigma, a_2b_2\sigma\}$

It is straightforward to illustrate that an XOR solution does not exist: let $h_1(\pi_1(a_1b_2))(\sigma) = 1$ and, thus, $h_2(\pi_2(a_1b_2))(\sigma) = 0$. Consequently, $h_2(\pi_2(a_1b_1))(\sigma) = 1$ and $h_1(a_1b_1)(\sigma) = 1$, leaving $h_1(\pi_1(a_2b_1))(\sigma) = 0$, thus resulting in a global enablement decision for σ of 0 after a_2b_2 occurs. We achieve the same result if instead $h_1(\pi_1(a_1b_2))(\sigma) = 0$. It is also straightforward to show that there is no XORSAT solution for this example. \diamond

3.3 Existence of controllers

It can be shown that decentralized controllers that correctly solve the control problem using \oplus as a fusion rule exist when K is XOR inference-observable.

Theorem 2. Given non-empty prefix-closed languages K , L such that $K \subseteq L$ and $\mathcal{L}(M_L) = L$. There exist n local decision functions h_1, \dots, h_n that form a valid global decision function H w.r.t. \oplus such that $\mathcal{L}(H/M_L) = K$ iff K is controllable and XOR inference-observable.

4. XOR OBSERVABILITY WHEN K IS DISTRIBUTED OBSERVABLE BUT NOT INFERENCE OBSERVABLE

When a language is not inference observable, the inferencing chain does not converge to a set of estimates either entirely in K or entirely in $L \setminus K$. Instead, the estimates start repeating from a previous link in the chain, thus forming a cycle of inferences. When a language is *not* distributed observable, it is straightforward to show the this cycle is of length 2; however, when a language is distributed observable but not inference observable, the cycles have different lengths, which we use to characterize when \oplus can be used as a fusion rule for this family of languages.

Definition 8. An *inferencing cycle* for $t\sigma \in L \setminus K$ w.r.t. $i_0 \in I_c(\sigma)$ is defined as follows.

$$(\exists t \in K)(\exists \sigma \in \Sigma) t\sigma \in L \setminus K \Rightarrow \quad (7)$$

$$(\exists k, m \in \mathbb{N} \text{ s.t. } k < m)(\exists i_0 \in I_c(\sigma)) \llbracket t \rrbracket_{i_0} \sigma \cap K \neq \emptyset \Rightarrow$$

$$\dots$$

$$(\exists i_k \in I_c(\sigma))(\exists t_k \sigma \in \llbracket t_{k-1} \rrbracket_{i_{k-1}} \sigma \cap K)$$

$$\llbracket t_k \rrbracket_{i_k} \sigma \cap L \setminus K \neq \emptyset \Rightarrow$$

$$\dots$$

$$(\exists i_{m-1} \in I_c(\sigma))(\exists t_{m-1} \sigma \in \llbracket t_{m-2} \rrbracket_{i_{m-2}} \sigma \cap K)$$

$$\llbracket t_{m-1} \rrbracket_{i_{m-1}} \sigma \cap L \setminus K \neq \emptyset \Rightarrow$$

$$(\exists i_m \in I_c(\sigma))(\exists t_m \sigma \in \llbracket t_{m-1} \rrbracket_{i_{m-1}} \sigma \cap L \setminus K)$$

$$\llbracket t_m \rrbracket_{i_m} \sigma = \llbracket t_k \rrbracket_{i_k} \sigma.$$

An *inferencing cycle* for $t\sigma \in K$ w.r.t. $i_0 \in I_c(\sigma)$ is similarly defined:

$$(\exists t \in K)(\exists \sigma \in \Sigma) t\sigma \in K \Rightarrow \quad (8)$$

$$(\exists k, m \in \mathbb{N} \text{ s.t. } k < m)(\exists i_0 \in I_c(\sigma)) \llbracket t \rrbracket_{i_0} \sigma \cap L \setminus K \neq \emptyset \Rightarrow$$

$$\dots$$

$$(\exists i_k \in I_c(\sigma))(\exists t_k \sigma \in \llbracket t_{k-1} \rrbracket_{i_{k-1}} \sigma \cap L \setminus K)$$

$$\llbracket t_k \rrbracket_{i_k} \sigma \cap K \neq \emptyset \Rightarrow$$

$$\dots$$

$$(\exists i_{m-1} \in I_c(\sigma))(\exists t_{m-1} \sigma \in \llbracket t_{m-2} \rrbracket_{i_{m-2}} \sigma \cap L \setminus K)$$

$$\llbracket t_{m-1} \rrbracket_{i_{m-1}} \sigma \cap K \neq \emptyset \Rightarrow$$

$$(\exists i_m \in I_c(\sigma))(\exists t_m \sigma \in \llbracket t_{m-1} \rrbracket_{i_{m-1}} \sigma \cap K)$$

$$\llbracket t_m \rrbracket_{i_m} \sigma = \llbracket t_k \rrbracket_{i_k} \sigma.$$

We denote a inferencing cycle w.r.t. $i_k \in I_c(\sigma)$, whether $t\sigma$ lies in $L \setminus K$ or K , by $i\text{-cycle}_{i_k}(t\sigma)$. The length of the cycle is defined by $|i\text{-cycle}_{i_k}(t\sigma)| = m - k$.

We can now identify whether or not a distributed-observable language that is not inference observable is XOR observable.

Definition 9. A distributed observable, but not inference observable, language $K \subseteq L$ is **XOR observable** if $(\forall t \in K)(\forall \sigma \in \Sigma_c) t\sigma \in L \setminus K \Rightarrow \exists i, j \in I_c(\sigma) \text{ s.t.}$

$$|i\text{-cycle}_i(t\sigma)| \bmod 4 = 0 \wedge |i\text{-cycle}_j(t\sigma)| \bmod 4 = 0.$$

Example 3. Let M_L and M_K be given as in Fig. 3. In this example, K is distributed observable but not inference-observable.

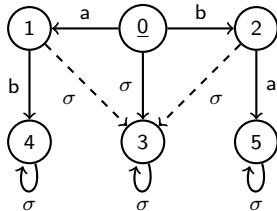


Fig. 3. Ex. 3 (adapted from Chakib and Khoumsi (2008): $\Sigma_{o,1} = \{a, \sigma\}$ and $\Sigma_{o,2} = \{b, \sigma\}$; $I_c(\sigma) = \{1, 2\}$).

It is straightforward to assert that K is distributed observable:

$$\llbracket a \rrbracket_1 \sigma \cap \llbracket a \rrbracket_2 \sigma \cap K = \emptyset; \llbracket b \rrbracket_1 \sigma \cap \llbracket b \rrbracket_2 \sigma \cap K = \emptyset.$$

We begin by trying to compute an inferencing chain w.r.t. controller 1 for the disablement decision for a w.r.t. σ :

- $\llbracket a \rrbracket_1 \sigma = \{a\sigma, ab\sigma, ba\sigma\}$
- $\llbracket ab \rrbracket_2 \sigma = \llbracket ba \rrbracket_2 \sigma = \{ab\sigma, ba\sigma, b\sigma\}$
- $\llbracket b \rrbracket_1 \sigma = \{b\sigma, \sigma\}$
- $\llbracket \varepsilon \rrbracket_2 \sigma = \{\sigma, a\sigma\}$
- $\llbracket a \rrbracket_1 \sigma = \{a\sigma, ab\sigma, ba\sigma\}$

We can see that the inferencing does not converge to a set of estimates either entirely in K or entirely in $L \setminus K$, but that it cycles back to the original set of estimates for $t\sigma = a\sigma$, i.e., $m = 4$ and $k = 0$, w.r.t. controller 1. We see similar results when beginning with controller 2:

- $\llbracket a \rrbracket_2 \sigma = \{a\sigma, \sigma\}$
- $\llbracket \varepsilon \rrbracket_1 \sigma = \{\sigma, b\sigma\}$
- $\llbracket b \rrbracket_2 \sigma = \{b\sigma, ab\sigma, ba\sigma\}$
- $\llbracket ab \rrbracket_1 \sigma = \llbracket ba \rrbracket_1 \sigma = \{ab\sigma, ba\sigma, a\sigma\}$
- $\llbracket a \rrbracket_2 \sigma = \{a\sigma, \sigma\}$

Thus, we have $i, j \in I_c(\sigma)$, i.e., $i = 1, j = 2$, such that their estimates of a produce inferencing cycles. It can be shown that inferencing cycles exist for the other disablement decision regarding σ for b . So we conclude that K is not inference-observable. In this case, all cycles have length 4, and, thus, K is XOR observable. A possible solution is shown in Table 5. \diamond

Table 5. Possible XOR solution for Ex. 3.

	$h_1(\cdot)(\sigma)$	$h_2(\cdot)(\sigma)$	\oplus	$H(\cdot)(\sigma)$
a	0	1	1	1
b	1	0	1	1
ε	1	1	0	0
ab	0	0	0	0
ba	0	0	0	0

4.1 Local control decisions

When K is not inference observable but is XOR observable, the local control decisions are defined using the same set of equations when K is inference observable. That is, for an inference cycle that begins with $t\sigma \in L \setminus K$, use Eq. (5) and repeat until the end of the cycle, otherwise use Eq. (6).

Revisiting Example 3: A set of local control decisions for Ex. 3 that produce a correct control solution under \oplus proceeds as follows. Begin with $i\text{-cycle}_1(a\sigma)$: because no local decisions have been assigned yet, we are free to choose a local decision and we can let $h_1(\pi_1(a))(\sigma) = 1$. As this is an anticipated global disablement decision, it must be the case that $h_2(\pi_2(a))(\sigma) = 0$. The next link in the cycle, according to Eq. (5) requires $h_2(\pi_2(ab))(\sigma) = 1$. As this is an anticipated global enablement decision under \oplus , we must have $h_1(\pi_1(ab))(\sigma) = 1$. The remaining local decisions for the links in the cycle are: $h_1(\pi_1(b))(\sigma) = 0$, $h_2(\pi_2(b))(\sigma) = 0$, and $h_2(\pi_2(\varepsilon))(\sigma) = 0$, $h_2(\pi_2(\varepsilon))(\sigma) = 0$. This is a different XOR solution than the one in Table 5, simply due to the fact that we initialized $h_1(\pi_1(a))(\sigma)$ to 1 and not 0.

4.2 Distributed observable but not inference observable and not XOR observable

For the family of distributed-observable languages that are neither inference observable nor XOR observable, there are no simple Boolean operations that will lead to a correct

control solution. Thus, in the absence of communication, there is no way to construct a solution that will be satisfactory using \vee , \wedge or \oplus (or their negation). Since K is still distributed observable, it is the case that if the controllers combined their information as per the definition of distributed observability, all controllers would be able to take the correct control decision; however, this effectively amounts to a broadcast. Instead, we recommend the synthesis of a point-to-point synchronous communication protocol (e.g., Ricker (2008)) to solve this class of languages.

Example 4. Let M_L and M_K be given as in Fig. 4. Here K is distributed observable, but is not XOR observable.

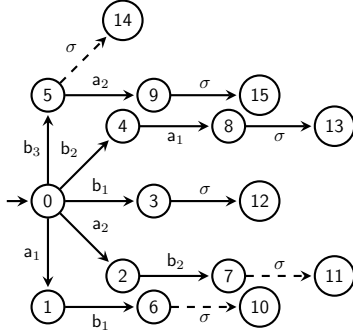


Fig. 4. Ex. 4: $\Sigma_{o,1} = \{a_1, a_2\}$, $\Sigma_{o,2} = \{b_1, b_2\}$, $I_c(\sigma) = \{1, 2\}$.

We examine the inferencing for one disablement decision w.r.t. controller 1:

- $\llbracket a_1 b_1 \rrbracket_1 \sigma = \{a_1 b_1 \sigma, b_2 a_1 \sigma\}$
 $\llbracket b_2 a_1 \rrbracket_2 \sigma = \{b_2 a_1 \sigma, a_2 b_2 \sigma\}$
 $\llbracket a_2 b_2 \rrbracket_1 \sigma = \{a_2 b_2 \sigma, b_3 a_2 \sigma\}$
 $\llbracket b_3 a_2 \rrbracket_2 \sigma = \{b_3 a_2 \sigma, b_3 \sigma\}$
 $\llbracket b_3 \rrbracket_1 \sigma = \{b_3 \sigma, b_1 \sigma\}$
 $\llbracket b_1 \rrbracket_2 \sigma = \{b_1 \sigma, a_1 b_1 \sigma\}$
 $\llbracket a_1 b_1 \rrbracket_1 \sigma = \{a_1 b_1 \sigma, b_2 a_1 \sigma\}$

And inferencing with the same sequence w.r.t. controller 2:

- $\llbracket a_1 b_1 \rrbracket_2 \sigma = \{a_1 b_1 \sigma, b_1 \sigma\}$
 $\llbracket b_1 \rrbracket_1 \sigma = \{b_1 \sigma, b_3 \sigma\}$
 $\llbracket b_3 \rrbracket_2 \sigma = \{b_3 \sigma, b_3 a_2 \sigma\}$
 $\llbracket b_3 a_2 \rrbracket_1 \sigma = \{b_3 a_2 \sigma, a_2 b_2 \sigma\}$
 $\llbracket a_2 b_2 \rrbracket_2 \sigma = \{a_2 b_2 \sigma, b_2 a_1 \sigma\}$
 $\llbracket b_2 a_1 \rrbracket_1 \sigma = \{b_2 a_1 \sigma, a_1 b_1 \sigma\}$
 $\llbracket a_1 b_1 \rrbracket_2 \sigma = \{a_1 b_1 \sigma, b_1 \sigma\}$

Note that these two sets of estimates both result in an inferencing cycle of length 6. It can be verified that this is also the case for the other two disablement decisions. As a result, we conclude that K is not XOR observable. It is another straightforward exercise to show that no XORSAT solution exists. \diamond

4.3 Existence of Controllers

It can also be shown that we can synthesize correct control decision functions that use \oplus as the fusion rule when K is XOR observable.

Theorem 3. Given non-empty prefix-closed languages K , L such that $K \subseteq L$ and $\mathcal{L}(M_L) = L$. There exist n local

decision functions h_1, \dots, h_n that form a valid global decision function H w.r.t. \oplus such that $\mathcal{L}(H/M_L) = K$ iff K is controllable and XOR observable.

5. CONCLUSION

We have provided a language-based characterization for the existence of parity-based architectures. It is unclear why \oplus might be preferred as the fusion rule for inference-observable languages over the existing \vee or \wedge , as constructing correct control solutions is somewhat more straightforward with these operators. But introducing \oplus for languages that are not inference-observable, but are distributed observable, allows us to extend the class of languages for which we can generate decentralized control solutions in the absence of communication between controllers.

REFERENCES

- Biere, Armin and Marijn J. H. Heule and Hans van Maaren and Toby Walsh. Handbook of Satisfiability. IOS Press, Vol. 185, 2009.
- Chakib, H. and Khoumsi, A. Multi-decision Supervisory Control: Parallel Decentralized Architectures Cooperating for Controlling Discrete Event Systems. *IEEE Trans. Automat. Control*, 56(11), pp. 2608–2622, 2011.
- Chakib, H. and Khoumsi, A. Multi-Decision Decentralized Control of Discrete Event Systems : Application to the C&P Architecture. In *Proc. 9th International Workshop on Discrete Event Systems*, 480–485, 2008.
- Kumar, R. and Takai, S. Inference-based ambiguity management in decentralized decision-making: Decentralized control of discrete event systems. *IEEE Trans. Autom. Control* 52(10) 1783–94, 2007.
- Prosser, J.H. and Kam, M. and Kwatny, H.G. Decision fusion and supervisor synthesis in decentralized discrete-event systems. In *Proc. Ameri. Contr. Conf.* pp 2251–2255, 1997.
- Qiu, W. and Kumar, R. and Chandra, V. Decentralized Control of Discrete Event Systems Using Prioritized Composition With Exclusion. *IEEE Trans. Autom. Control* 53(10): 2425 – 2430, 2008.
- Ramadge, P. and W.M. Wonham Supervisory Control of a Class of Discrete Event Processes. *SIAM J. Contr. Optim.* 25: 206–230, 1987.
- Ricker, S.L. and H. Marchand A parity-based architecture for decentralized discrete-event control. In *Proc. Ameri. Contr. Conf.*, pp. 5678–5684, 2013.
- Ricker, S. L. Asymptotic Minimal Communication for Decentralized Discrete-Event Control. In *Proc. 9th International Workshop on Discrete Event Systems*, pp. 486–491, 2008.
- Ricker, S. L. and K. Rudie, Know Means No: Incorporating knowledge into discrete-event control systems. *IEEE Trans. Autom. Control* 45(9):1656–1668, 2000.
- Rudie, K. and Wonham, W.M. Think Globally, Act Locally: Decentralized Supervisory Control. *IEEE Trans. Autom. Control* 37:1692–1708, 1992.
- Yoo, T.-S. and S. Lafortune A general architecture for decentralized supervisory control of discrete-event system. *Discrete Event Dyn. S.*12: 335–377, 2002.
- Yoo, T.-S. and Lafortune, S. Decentralized supervisory control with conditional decisions: supervisor existence *IEEE Trans. Autom. Control* 49(11):1886–1904, 2004.